DEPARTMENT OF ARTIFICIAL INTELLIGENCE & MACHINE LEARNING
ST JOSEPH ENGINEERING COLLEGE, Mangaluru
2022-2023

Synopsis and Design Phase Report

On

FAMILY DATABASE MANAGEMENT SYSTEM

SUBMITTED BY

VIJNA SHETTY **4SO20AI058**

VINEETH SHENOY P **4SO20AI059**

01/12/2022

Table of Contents

PREFACE

This is the report of Family Database Management system designed by Vineeth Shenoy P and Vijna Shetty of Department of ICBS. The Report contains the Problem statement, Requirements Specifications, Design of solution, Software development Model used and its implementation, Testing phase details, software and framework used for developing the Backend and Frontend of the Application. As of the current Edition (edition 1), the report will consist the title of the project, Problem statement, Detailed Description of the project, Software requirements specification and the Design of the Database. The design of the database includes the specification of various entities identified, their relationships, attributes, constraints, their representation in an ER diagram and Schema diagram. The future editions will contain various updates and the changes compared to previous editions will be specified in every preface.

1.PROJECT TITLE: FAMILY DATABASE MANAGEMENT SYSTEM

2.PROBLEM STATEMENT:

In the current generation, we find it difficult to remember the names of relatives further than 1 generation. Every time we see our far away relatives, we are not able to recognize who that person is and what is their relationship with us.

We forget the birthday and wedding anniversary of relatives who are far away from us, both by distance and generation.

We forget their address and phone number; it is always difficult to find the contact number on our phone if there is someone else with the same name.

Our Family Database Management System will resolve all of these problems and provide many more features.

3. DETAILED DESCRIPTION OF PROJECT:

The detailed description of the project can be given as follows:

- Our Project will be developed using the "Waterfall Model" of software development. There will be changes in the requirements specification and design of our project specified in the current report. Any changes in the Requirements and design that may have been reflected during the implementation and testing phases will be specified in future editions of the report.
- This Database Management System will be implemented as a website and will be open for all users, if they register.
- This Database Management System will store the data of every family member and trace the relationship with the family member when requested for.
- It will return the name of the relation (E.g.: Cousin/Brother-in-law etc.,) and display a tree representation of the relationship between any family member to another family member, if and only if a relationship exists.
- It will have event reminder systems, birthday reminders, Wedding Anniversary reminder, and contact retrieval.
- Event reminders can invite and remind specific families or couples.
- It will return the address, phone number, and other personal details based on constraints, whenever requested for.
- It will maintain the count of members in each family.

- In future updates it will be utilizing image recognition features to recognise family members and return their name and relationship.

4. SOFTWARE REQUIREMENT SPECIFICATION:

The Requirements of a system are the descriptions of the services that a system should provide and the constraints on its operation. These requirements reflect the need of customers for a system that serves a certain purpose such as controlling a device, placing an order, or finding information.

The Family Database Management System has its requirements specification too. We can classify our requirements as User and System.

4.1. User Requirements:

User requirements are the statements in natural language and explains what are the services expected from the system. The users of our system will be anyone who wishes to store their family members data onto the database and retrieve their personal data and relationship with other family members when requested. Keeping that in mind we can specify the user requirements as:

- The system must be able to store the data of every family member.
- If we enter the name of 2 family members, the system must be able to tell the user what is the relationship between them.
- The system must be able to display the above-mentioned relationship in the form of a tree.
- When we enter the details of an event and the invitees for it, the system must be able to send the invites to all the family members listed in "invitees."
- The system must be able to send reminders of birthdays of family members to everyone else on the right day.
- The system must be able to send reminders of wedding anniversary of a couple to all other family members.
- The system must be able to keep a track of the count of the family members in each family.

We can explain the requirements in much more detail with the system specifications and requirements.

4.2. System Requirements and Specifications:

System or Software requirements are a much more detailed specifications of user requirements but with a software developers' point of view. It will give a deeper understanding of what the system is supposed to do and are classified as functional and non-functional requirements. The functional and non-functional requirements of our system are specified as follows

4.2.1. Functional Requirements:

These are statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situations. In some cases, the functional requirements may also explicitly state what the system should not do.

The functional requirements of Family Database Management System are:

- The system must store the Date of Birth, Name, first name and last name of the user/family member.
- It must generate a unique ID based on the full name and date of birth of the family member. Any collisions detected while creating the ID must be resolved by generating another unique ID.
- When the User enters the name of 2 relatives, the system must use a precise and efficient algorithm to trace the relationship between the family members.
- The relationship must be in English such that, it must give the precise name of the relationship. For Example, the system must be able to specify whether a relationship is Maternal or Paternal.
- When the user requests the system to display the family tree between the 2 members, the system must display the tree with the 2 members as the start and end node respectively. All other family members responsible for the relationship must be displayed in the form of intermediate nodes.
- When the user enters his/her personal details such as phone number, address, job details, etc., the system must save the data and must display the data to authorized users only. Phone number and address can be displayed to all users, whether authorized or not.
- The system must remind the birthday of a family member to all other family members, if and only if they are related to each other. The same constraint applies for Wedding anniversary reminders.
- The event reminders will be used to invite and remind family of an event and its date. The constraint for inviting will be specified by the user.
- The constraint for the invitation is the invitee list specified by the user. If all the members of a family must be invited then the system must invite the members of that family only. If the user wants to invite only a certain set of members based on their relationship with the couple, then the system will invite only those couples.

4.2.2. Non-Functional Requirements:

Non-functional requirements are the requirements that are not directly concerned with the specific services delivered by the system to its users. They may relate to emergent system properties such as reliability, response time, and memory use. They may also define constraints on the system implementation, such as the capabilities of I/O devices or the data representations used in interfaces with other systems. They are much more critical.

The Non-Functional requirements of Family Database Management System are:

- The System must be deployed as a webpage available to all users.
- It must be a secure website; in technical terms it must be https. This is mainly due to the system's feature to store the personal details of family members and these details cannot be disclosed to unauthorized users.
- It must have a good response time. The system must return the relationship value within few seconds.
- The entire UI of the webpage must be smooth for the users.
- The server that would store the database must have sufficient storage.

As we are not exposed to a development environment, it is difficult for us to specify the non-functional requirements. They will be recognised as we develop the software.

4.2.3 Software and Hardware Required:

- Backend: Django, Python 3.10
- Frontend: HTML, CSS, JS
- Database Language: MySQL
- IDE: VS Code
- Repository Server: Github
- UML Diagram Design Software: Lucidchart
- System Requirement:
    - Processor: i3 or above,
    - RAM: 4GB or above,
    - Browser: Brave, Edge, Chrome.

5. Design of Database:

5.1 Entities, attributes, and relationship:

- Family Member (ID, FName, Name, LName, DoB, DoD)
- Couple Family (Couple_ID, Hus_ID, Wife_ID, Wed_Ann)
- Parents (ID, Father_ID, Mother_ID)
- Personal Info (ID, Ph_Prefix, Phone, Address, Job_Desig, Company, Privilege)
- Families (Fam_ID_Code, Fam_Name, Members)
- Events (Event_Name, Event_ Date, Event_Code)
- Invitees (Event_Code, Family_Invited, Couples_Invited)

5.2 Table Descriptions:

5.2.1 Family_Member:

Field	Type	Null	Key	Default
ID	VARCHAR(18)	NO	PRI	NULL
FName	VARCHAR(20)	NO		NULL
Name	VARCHAR(20)	NO		NULL
LName	VARCHAR(20)	NO		NULL
DoB	DATE	NO		NULL
DoD	DATE	YES		NULL

5.2.2 Couple_Family:

Field	Type	Null	Key	Default
Couple_ID	VARCHAR(18)	NO	PRI	NULL
Wed_Ann	DATE	NO		NULL
Hus_ID	VARCHAR(18)	NO	MUL	NULL
Wife_ID	VARCHAR(18)	NO	MUL	NULL

5.2.3 Parents:

Field	Type	Null	Key	Default
ID	VARCHAR(18)	NO	MUL	NULL
Father_ID	VARCHAR(18)	NO	MUL	NULL
Mother_ID	VARCHAR(18)	NO	MUL	NULL

5.2.4 Personal_Info:

Field	Type	Null	Key	Default
ID	VARCHAR(18)	NO	PRI	NULL
Ph_Prefix	INT	NO		NULL
Phone	BIGINT	NO		NULL
Address	VARCHAR(50)	YES		NULL
Job_Desig	VARCHAR(20)	YES		NULL
Company	VARCHAR(20)	YES		NULL
Privilege	TINYINT(1)	NO		NULL

5.2.5 Families:

Field	Type	Null	Key	Default
Fam_ID_Code	VARCHAR(18)	NO	PRI	NULL
Fam_Name	VARCHAR(20)	NO		NULL
Members	INT	YES		NULL

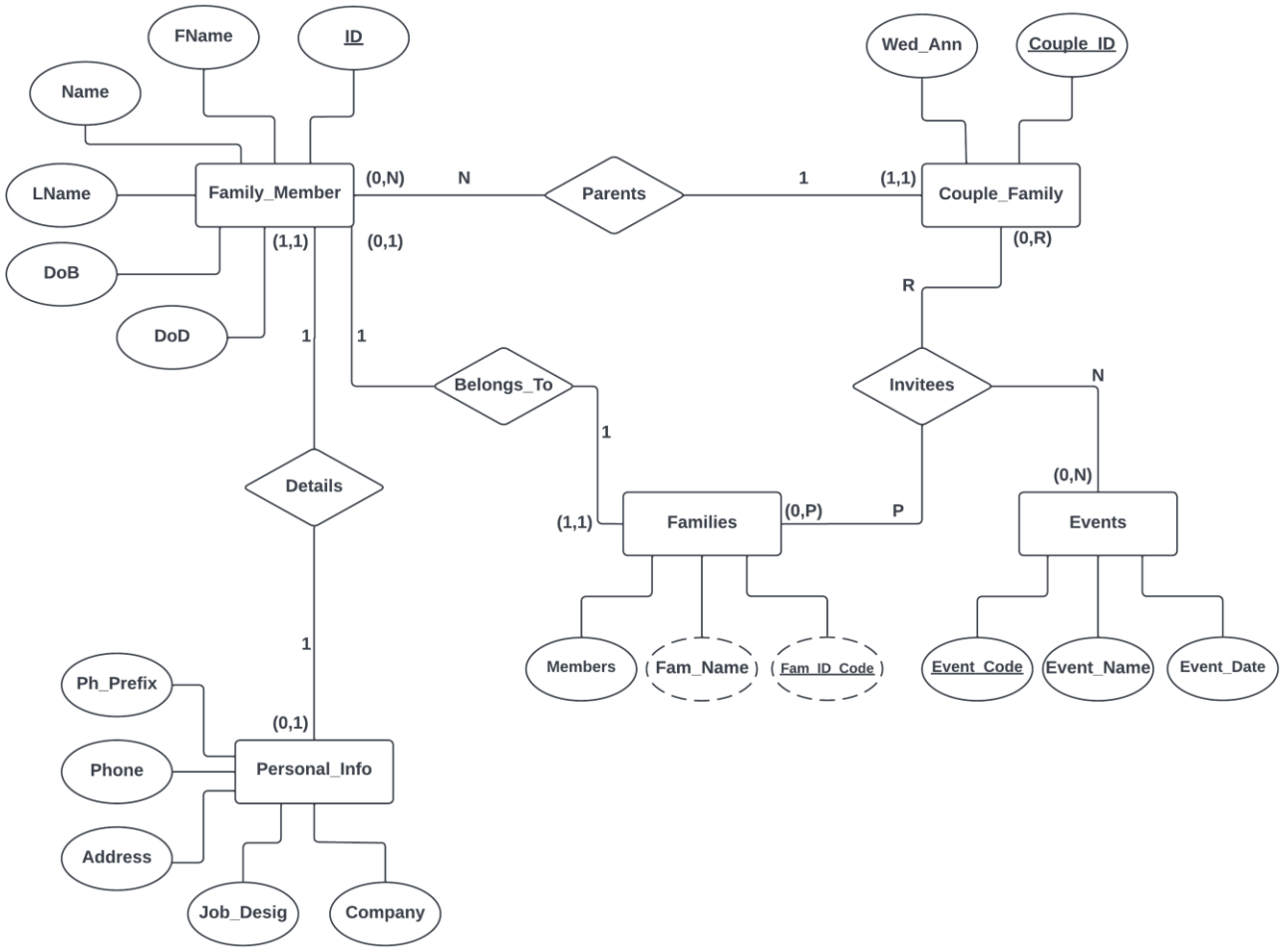
5.2.6 Events:

Field	Type	Null	Key	Default
Event_Code	VARCHAR(18)	NO	PRI	NULL
Event_Name	VARCHAR(20)	NO		NULL
Event_Date	DATE	NO		NULL

5.2.7 Invitees:

Field	Type	Null	Key	Default
Event_Code	VARCHAR(18)	NO	PRI	NULL
Family_Invited	VARCHAR(18)	YES	MUL	NULL
Couples_Invited	VARCHAR(18)	YES	MUL	NULL

5.3 Entity-Relationship Diagram:



5.4 Schema Diagram: